

Prime number

$17 \Rightarrow 2, 3, 5, 7, 11, 13 \rightarrow \text{loop}(2 \text{ to } n-1)$

↓
not divisible by any
↓
Prime number

Prime number function

```
class Solution {
    public int countPrimes(int n) {
        int count = 0;
        for(int i=2; i<n; i++){
            if(isPrime(i) == true){
                count++;
            }
        }
        return count;
    }

    public static boolean isPrime(int n){
        for(int i=2; i<n; i++){
            if(n%i==0){
                return false;
            }
        }
        return true;
    }
}
```

TLE = ?

compiler → is $\Rightarrow 10^8$ instructions

$n = 10^6$

$n^2 = 10^6 \cdot 10^6 = 10^{12}$ instructions

↙
TLE

$N = a \times b$

$$\begin{cases} N \rightarrow \text{prime} \\ a, b = 1, N \end{cases}$$

} a, b ≠ factors

$N = a \times b$

$N \rightarrow \text{not prime}$

$$\begin{array}{ccc} a = b & a > b & a < b \\ \downarrow & \downarrow & \downarrow \\ a & a & b \end{array}$$

$N = a \cdot a$

$a \uparrow$

$N = a^2$

$b \uparrow$

$a = \sqrt{N}$

$b < \sqrt{N}$

$a < \sqrt{N}$

one of the
factors will be
found till \sqrt{N}

$81 \Rightarrow \sqrt{81} = 9$

$27 \times 3 = 81$

$1, 3, 9, \cancel{27}$

$32 \Rightarrow 1, 2, 4, \cancel{8}, \cancel{16}, \cancel{32}$

$\sqrt{32} = 5 + 0.6$

$i < \sqrt{n}$

\downarrow

$$\begin{array}{l} i^2 < n \\ i+i < n \end{array}$$

some inequalities


```

class Solution {
    public int countPrimes(int n) {
        int count = 0;
        for(int i=2; i<n; i++){
            if(isPrime(i) == true){
                count++;
            }
        }
        return count;
    }

    public static boolean isPrime(int n){
        for(int i=2; i*i <= n; i++){
            if(n%i==0){
                return false;
            }
        }
        return true;
    }
}

```

Time = $n \cdot \sqrt{n}$

$n = 10^6$

$$6^6 \cdot \sqrt{10^6}$$

$$10^6 \cdot (10^6)$$

16⁶ . 10^{6/2}

$$10^6 \cdot 10^3 = 10^9$$

TLE

Sieve of Eratosthenes

$$\frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7}$$

$$\sqrt{100} = 11$$

$$\Rightarrow n \left[\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \dots \right]$$

71

०८८

$$n \cdot \log_2(\log_2 N) \quad n = 1024 = 2^{10} \quad \log_a^b = b \cdot \log_a \\ \log_a a = 1$$

$$2^{10} \cdot \log_2(\log_2 2^{10})$$

$$2^{10} \cdot \log_2(18 \cdot \cancel{\log_2 2})$$

$$2^{10} \cdot \log_{10} 10$$

n. 3.17 ~

[View Details](#)

Paliadromic substrings

Variations

11 - Count all PCL substrings

21. Find longest pal sub.

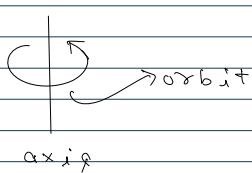
31- count all odd-length pal sub-

en - even

51. Find the longest odd - .

61. Cireh

Copyright © The McGraw-Hill Companies, Inc.

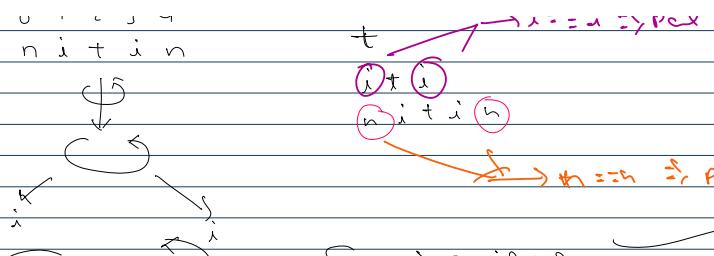


~~Fixing Orbit Approach~~
Expand around center appr oach

expand around center approach

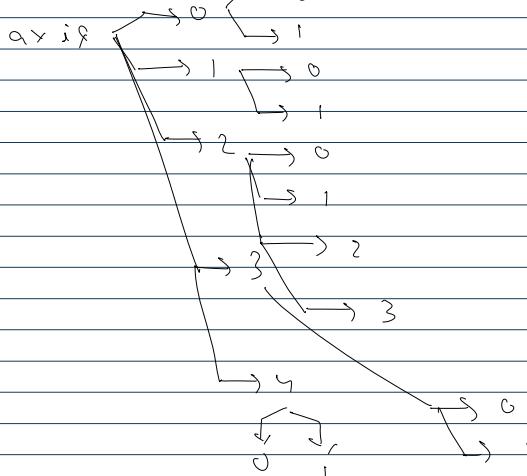
o i 2] 4
n i t i n

$i = i \Rightarrow p \in L$



Odd length palindromes

0 1 2 3 4 n i t i n	0 1 2 3 4 n i t i n	0 1 2 3 4 n i t i n	0 1 2 3 4 n i t i n	0 1 2 3 4 n i t i n
\downarrow axis = 0	\downarrow axis = 1	\downarrow axis = 2	\downarrow axis = 3	\downarrow axis = 4
$0 - 0 = 0 \quad \{ n$ $0 + 0 = 0$	$1 - 0 = 1 \quad \{ \lambda$ $1 + 0 = 1$	$2 - 0 = 2 \quad \{ +$ $2 + 0 = 2$	$3 - 0 = 3 \quad \{ \lambda$ $3 + 0 = 3$	$4 - 0 = 4 \quad \{ \lambda$ $4 + 0 = 4$
$0 - 1 = -1 \quad \times$ $0 + 1 = 1$	$1 - 1 = 0 \quad \{ \text{nit} \times$ $1 + 1 = 2 \quad \{ \text{nit} \times$	$2 - 1 = 1 \quad \{ i = i$ $2 + 1 = 3 \quad \{ i + i$	$3 - 1 = 2 \quad \{ + f n$ $3 + 1 = 4 \quad \times$	$4 - 1 = 3 \quad \{$ $4 + 1 = 5 \quad \circ$
Gorbit	$\text{Orbit} = 1$	$\text{Orbit} = 1$	$\text{Orbit} = 1$	$\text{Orbit} = 1$
axis-orbit	axis+orbit	axis-orbit	axis+orbit	axis-orbit
$2 - 3 = -1 \quad \times$ $2 + 3 = 5$				



```

public class Main {
    public static void main(String[] args) {
        String s = "nitin";
        System.out.println(palindromicSubstringCount(s));
    }

    public static int palindromicSubstringCount(String s){
        // Odd length
        int odd = 0;
        for(int axis=0; axis<s.length(); axis++){
            String temp = "";

```

5 4

3

X

```

for(int orbit=0; axis-orbit >= 0 && axis+orbit < s.length() ; orbit++){
    if(s.charAt(axis-orbit) == s.charAt(axis+orbit)){
        odd++;
        if(axis-orbit != axis+orbit){
            temp = s.charAt(axis-orbit) + temp + s.charAt(axis+orbit);
        }
        else{
            temp = "" + s.charAt(axis-orbit);
        }
        System.out.println(temp);
    }
}
}
}
return odd;
}
}

```

Even length substring

$$\left\{ \begin{array}{l} \text{axis-orbit} \rightarrow 0.5 \\ \text{axis+orbit} \rightarrow 0.5 \end{array} \right.$$

o 1 2 3
n a a n
↓
axis = 0.5

o 1 2 3
n a a n
↓
axis = 1.5

o 1 2 3
n a a n
↓
axis = 2.5

o 1 2 3
n a a n
↓
axis = 3.5

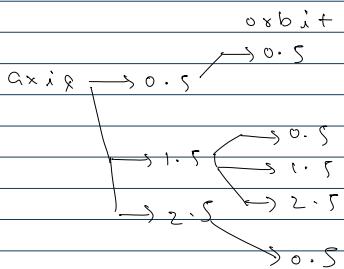
$$0.5 - 0.5 = 0 \quad \{ \text{nf} a \\ 0.5 + 0.5 = 1 \quad X$$

$$1.5 - 0.5 = 1 \quad \{ a = a \\ 1.5 + 0.5 = 2 \quad \{ \text{aa}$$

$$2.5 - 0.5 = 2 \quad \{ a^2 n \\ 2.5 + 0.5 = 3 \quad X$$

$$1.5 - 1.5 = 0 \quad \{ n = n \\ 1.5 + 1.5 = 3 \quad \{ \text{naan}$$

$$1.5 - 2.5 = -1 \quad \{ X \\ 1.5 + 2.5 = 4$$



merge 2 sorted array

arry1 → 2, 3, 5

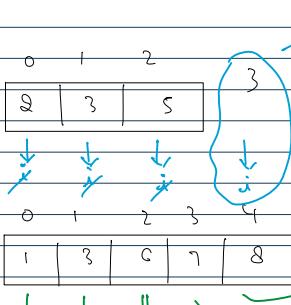
arry2 → 1, 3, 6, 7, 8

arry → 1, 2, 3, 3, 5, 6, 7, 8

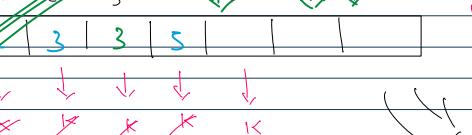
2, 3, 5, 1, 3, 6, 7, 8

Bubble
sort $\sim n^2$
 $\sim n \log n$
merge

1, 2, 3, 3, 5, 6, 7, 8



arry1 complete



i = 0
j = 0
k = 0

arry

$\text{arry1}[i] < \text{arry2}[j] \Rightarrow \text{arry1}[i] = i, j++$, k++
 $\text{arry1}[i] > \text{arry2}[j] \Rightarrow \text{arry2}[j] = j, i++$, k++

```

public class Main {
    public static void main(String[] args) {

```



```
int[] arr1 = {2, 3, 5};  
int[] arr2 = {1, 3, 6, 7, 8};  
  
int[] arr = merge(arr1, arr2);  
  
for(int i=0; i<arr.length; i++){  
    System.out.print(arr[i] + " ");  
}  
}  
  
public static int[] merge(int[] arr1, int[] arr2){  
    int n = arr1.length;  
    int m = arr2.length;  
  
    int[] arr = new int[n+m];  
  
    int i = 0;  
    int j = 0;  
    int k = 0;  
  
    while(i<n && j<m){  
        if(arr1[i] < arr2[j]){  
            arr[k] = arr1[i];  
            i++;  
            k++;  
        }  
        else{  
            arr[k] = arr2[j];  
            j++;  
            k++;  
        }  
    }  
    while(i<n){  
        arr[k] = arr1[i];  
        i++;  
        k++;  
    }  
  
    while(j<m){  
        arr[k] = arr2[j];  
        j++;  
        k++;  
    }  
  
    return arr;  
}
```

